Sept 1, 2009.

This document was produced by the baseline working group. While it marked with draft status, it was agreed by Mark Leiniger, acting for the CIO, and Mark Kaletka, that if would form the basis for tooling for the baseline verification and enforcement process, and be the basis for organizing baseline work.

The intent is to further develop the document, given some initial experience with the tools.

-- Don Petravick

Fermilab Baseline Process Framework

Joe Boyd, Matt Crawford, Irwin Gaines, Jamie Blowers, Don Petravick, Connie Sieh, Kirk Skaar, Tim Zingelman

1 Baseline Process Scope

All Fermilab IT systems are subject to configuration management, as mandated in the Fermilab Policy on Computing. Major configuration items will have lab standard baselines that go through each of the steps described here; minor elements that exist in only small quantities and do not carry significant risks with them can be managed more informally based on locally defined configuration processes. This document describes the procedures for determining which configuration elements are classed as major or minor, and gives the details of the processes used to manage baselines for major configuration elements.

The software and systems allowed at Fermilab are a function of Fermilab's Enterprise architecture, and are not in scope of this document.

Concerns

This section describes how CIO concerns determine which systems require formal baselines, how systems are organized into classes that share a common baseline, and how major elements are managed by using common baselines and individual variances.

A **concern** is a characteristic of IT systems that the CIO is accountable for controlling across all Fermilab IT. Currently, the CIO concerns include **Security,** and **Environmental Impact**. Other aspects of systems operation are overseen, but not controlled by the CIO. Systems that are the object of such concerns need to be managed according to the formal baseline processes laid out in this document. As guidance, systems having configurable elements affecting one or more of the CIO's concerns and meeting one or more of the following criteria are likely to require full baselines:

- Systems used by more than one division, section or center;
- Systems having more than 10 instances overall.

Generalizing from the Fermilab Computer Security Plans, controls addressing the CIO's

concerns are effected by mandatory configurations for the relevant configurable elements of computers. Examples of configurable elements are: an operating system setting; an application setting; a required patch level; a sticker applied to a machine. Other configurable elements on the same systems (for example, the color of the computer or the size of the monitor) can be left up to the needs of the user and not regulated by baselines.

A particular configurable element falling into a category of concern can be of two types. G**lobal elements** are managed by a specification of behavior ALL baselined systems MUST have. S**tatistical elements** are managed by specifications requiring that some fraction of systems meet a goal. Both types of elements are listed as requirements in the baseline documents; the difference in how they are handled lies in the variance process described below.

Classes of Systems

For our purposes, the CIO lumps operating system configurations into **classes,** so that a baseline may be written for the class, and so that application baselines may refer to the classes. Classes are such that, at a given time, a physical or virtual computer is never in more than one class and all baselined physical or virtual computers are in some class. A class is identified by CIO-defined attributes. There are three types of attributes: Operating System, Context and Enclave. Classes can be modeled with a matrix:

	Linux 3-6	OSX 10.5	XP	Vista	Sever 2003	Server 2008	Solaris 2.8, 2.	Solaris 10
Desktop/laptop, GCE, Internal network	x	x	x	x				
Server, GCE, Internal network	x	x	x	x	x	x	x	x
Server, OSE	x							

	Linux 3-6	OS X 10.5	XP	Vista	Server 2003	Server 2008	Solaris 2.8-2.9	Solaris 10	VxWorks
Desktop/Laptop, GCE, Internal network	X	X	X	X					
Server, GCE, Internal network	X	X	X	X	X	X	X	X	L
Server, OSE	X								

Figure: OS class matrix at the time version one of the document was written.

an X indicates that the class has a baseline. An L indicates that systems of this class rely on **locally defined configuration processes**. A blank indicates that no systems of this class are expected to exist. A single baseline document can cover multiple classes. Baseline documents may incorporate other baseline documents -- For example, at the

time of this writing, the GCE linux desktop baseline document incorporates the "Common Unix Baseline" document

All baselined computing systems will have on record which class they belong to so that it is known which baseline applies to that system.

In this document we do not assume that one organizational entity manages every computer in a class.

Baselines and Variances

A **baseline** is a document, approved by a formal process and under change control, that specifies the Global and Statistical configurable elements and their required values. A given baseline applies to one or more **classes of computers**. A **variance** specifies alternate controls, approved by a formal process with change control, that is granted to computers that cannot comply with one or more specified configuration elements..

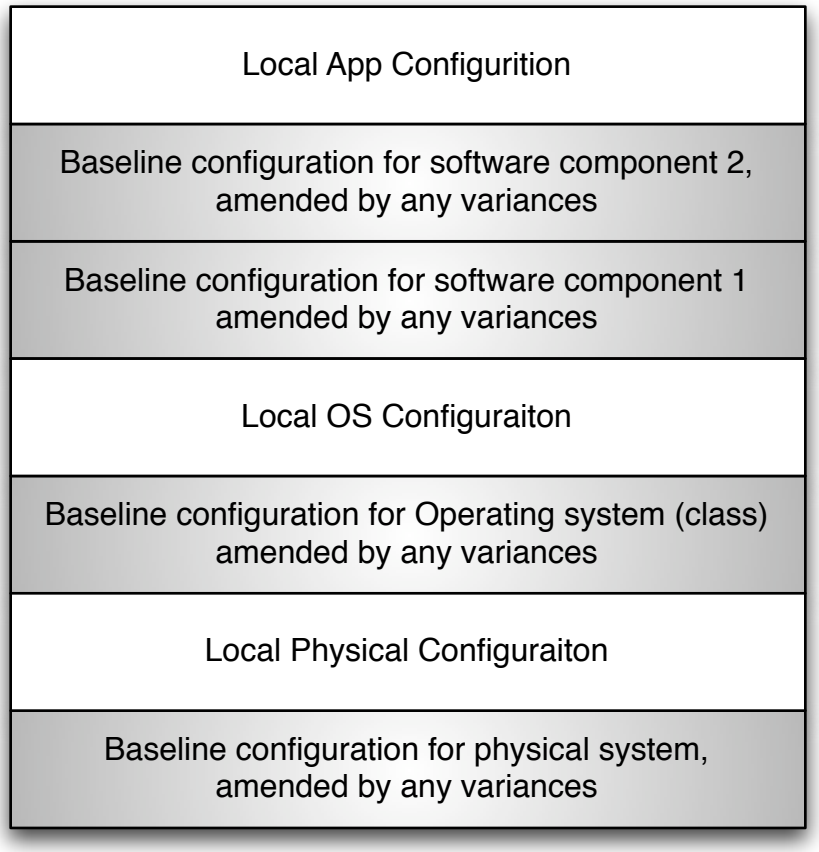
System baselines represent the highest level in a hierarchy of baselines that apply to specific systems. For example, there may be a system baseline for Linux operating systems that applies to all classes of computers running Linux. A system belonging to the class of SL5 systems will have additional baseline requirements that apply specifically to SL5. For a given concern, every deployable class is covered by a unique baseline hierarchy (of system and specific baselines), and never covered by two.

Some software components are heavily reused at Fermilab. Examples are the Apache Web server, and various DBMS systems. It is economical to have distinct baselines for components in wide use at Fermilab. Like common systems baselines, such **component baselines** are valid for the **classes of computers** for which they are written. The CIO keeps a **list** of **components of concern**, and specifies classes of computers where the components might be deployed.

Baselines and policies

Baselines are not the only source of constraints on system configuration: Certain computer security policies put limits on systems' network behavior and use, independent of hardware and software details. Security policies govern system behavior without specifying implementation; a configuration baseline specifies implementation, and will often go beyond the security policies.

1.1 Summary of Baselines and Configuration Management



The figure above illustrates the configuration of a typical computer at Fermilab. Elements of its configuration (shaded) are controlled by various baselines, possible modified by variances. All other elements of its configuration are controlled by local processes.

2 CIO's governance of all Baselines

This process:

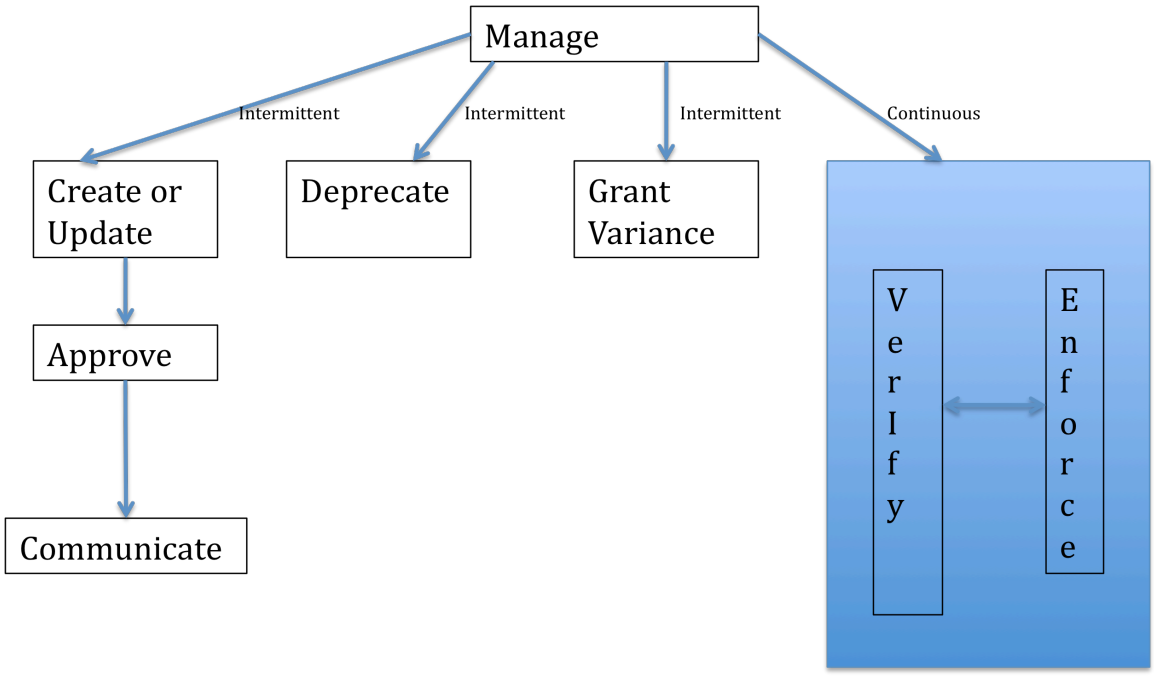
- Determines the number and kind of common system baselines and component baselines.
- The relationship of baselines to each other.
- Authorizes the construction/update and deprecation of baselines.
- Determines the organizational unit responsible for managing the baseline.
- Maintains this document.
- Reviews the approvals of baselines, baseline updates, and variances.

- Provides guidance to baseline projects in the areas of law, regulation, lab contract, and other external constraints.

Having determined that a particular CE requires a baseline, responsibility for composing that baseline will be assigned to a particular lab organization. Guidance for this assignment is as follows: In cases where the CE is used by multiple lab Divisions/Sections/Centers (DSCs), the Computing Division will usually be responsible for the baseline. In cases where only a single DSC uses the CE, that DSC will be responsible.

A baseline is deprecated when the CIO has no confidence that it is suitable for its purpose, and the CIO deems that the baseline will not be updated. Any remaining CE's are then governed under local configuration management processes.

3 Processes for a Single Baseline



3.1 Manage a Baseline

Each baseline has a unique organizational unit that manages the baseline. The head of that unit is accountable to the CIO for managing the baseline. The head of the OU typically names a manager responsible for the baseline, in consultation with the CIO. The community is informed of the manager selected.

The work of baseline management includes:

- Monitoring that the baseline achieves its purpose.
- Monitoring internal and external triggers indicating a need to update or deprecate the baseline, informing the CIO as needed.
- Running the continuous baseline lifecycle processes.
- As needed, Initiating and running the non-continuous baseline lifecycle processes.

3.2 Compose/Update a Baseline

3.2.1 Deliverables

This process can create an initial release of a baseline or a new version of an existing baseline. This process produces a **release** of

- Baseline document
- Baseline instrumentation artifacts
- Baseline impact assessment.

The baseline shall be published in PDF format. Draft documents shall be stored separately from approved documents.

The instrumentation artifacts include those used in the **enforce**, **announce** and **verify** processes, described below.

Impact states the burden on the community to implement the baseline (for example, "it will require N person-hours and a system downtime of M days to bring our systems into compliance with the proposed baseline")

3.2.2 Process

Like all ITIL release processes, this process considers all technical and non-technical aspects of a release by taking a holistic view of implementing changes to IT services, including the needs of all systems to which the baseline applies.

Experience has shown that the amount of work depends on the diversity of the systems and communities relying upon them. The create/update typically includes the following steps:

- Baseline manager names a leader who is accountable for the process and result.
- The Leader makes suitable announcements to the community that a baseline creation (or revision) process is underway.
- A person of appropriate organizational stature drafts a team.
- Team is briefed by those responsible for interpreting applicable external constraints e.g. law, regulations, or contract.

- Team provides communication plan (e.g. Comments should be sent to a "user" email list requesting feedback)
- Team drafts the proposed baseline
- Team reviews variances from the baseline, and revises them as apropos.
- Team produces associated technical artifacts.
- Team prepares an impact assessment for the proposed baseline, in consultation with the user community and the CIO's experts.
- The team deals with user feedback per the previously published communication plan.
- Submits results to approval OU manager.

3.2.3 Configuration Items

Baselines SHALL use key words described in RFC 2119. A baseline consists of descriptions of a collection of "settings" of hardware and software parameters for the CE in question. Only required settings will be included; optional or recommended settings will be documented separately.

The mandatory settings under discussion here are referred to here as "configuration items." Each configuration item in a baseline must include each of the following attributes.

- A unique invariant identifier
- Name of the configuration item
- Clear description of what the setting consists of.
- Justification
- Required setting
- How to verify compliance
- How to become compliant
- Non-compliance action
- Length of period of remediation

Notes:

Justification consists of naming the concern(s) addressed by the configuration item, e.g. "security".

The configuration baseline may specify attributes that are not directly specified in the baseline itself, but incorporated by reference. Such attributes are called **indirect**. Such a reference might be made, for example to establish what the latest patches for an application are. E.g. in the spirit of "set this parameter to the value given on the Linux system configuration web page".

The non-compliance action and the period of remediation are used by the baseline enforcement process, specified below. Normally a period of remediation would

range from 1 day to 1 month. A typical compliance action is to deny network service to the computer. Periods of remediation can be zero, in which case the action is taken immediately, this is used for items which represent an immediate danger if mis-configured.

3.3 Approve a Baseline

Newly created or updated baselines must be approved by head of the OU managing the baseline.

This indicates

- The head's concurrence that the proposed baseline provides appropriate mediation of identified concerns,
- The head's understanding of the work required to manage the baseline,
- The head's agreement that the impact of the baseline is appropriate.

After the head's approval, CIO is notified. The baseline goes into effect after either the CIO concurs, or passage of given sufficient time for CIO to raise objections, but no objections received, normally 2 weeks, or a period agreed to by the CIO.

3.4 Communicate a Baseline

While each system administrator is responsible for knowing the contents of the repository, there is an announcement process indicating to the community that a new document is in the repository, currently CD DocDB. Baseline documents shall be available to all Fermilab people with a need to know. Any transmissions of baselines to DOE or other outside entities will only come from the CSEXEC

Baselines shall be published in pdf format.

The approval status, version number, revision date and managing OU shall be clearly stated.

3.5 Deprecate a Baseline

This process:

- Announces the deprecation to the community.
- Waits for a period of time.
- Tears down baseline management artifacts.

- Marks baseline documents as deprecated

Note:
Any systems not covered by some baseline after deprecation revert to locally defined configuration processes.

4 Grant Variance

In some circumstances a CE may not be able to be brought into compliance with one or more elements of the relevant baseline. In such cases the system owner can request a variance from the baseline. Variances are recorded.

To review, there are two types of **concerns** that are controlled by baselines. One type of concern, the **global concern,** results in a specification of behavior ALL baselined systems must have. The other type of concern, the **statistical concern,** results in specifications such that some fraction of systems meet a goal.

Variances are granted for a duration and are reviewed before the end of the specified duration. Regardless of duration, variances are reviewed, and possibly adjusted, when the corresponding baseline is updated.

Variances detail specific configuration items in the baseline that will not be applied and any additional configuration items, not specified in the baseline, that compensate for the omitted configuration items. New configuration items are specified with the same number and kind of attributes as in the corresponding baseline, allowing them to fit into the **baseline verification process** detailed below. Variances from statistical concerns will typically relax the corresponding configuration settings, if warranted by the current level of compliance.

Initial variance requests are made to the manager of the baseline. In unusual cases, the manager may grant relief beyond the period of remediation specified in the baseline.

A full variance request includes management signoff by the OU responsible for the system requesting a variance. For each baseline element from which a variance is requested the following documentation must be provided:

- The reason the system cannot be brought into compliance with the baseline.
- What will be done instead of the baseline setting. For example, the request must describe what configuration will be applied instead of the baseline configuration.
- An explanation of why the alternative will appropriately address the concern covered by the baseline element.

After appropriate consultations, likely involving support groups and the CIO's designated expert for the concern, the baseline manager approves or denies the variance.

5 Continuous Baseline Processes

The continuous baseline processes provide assurance that the specified baselines are actually deployed in the field. The machinery used by these processes may also be used by other processes. For example the critical vulnerability process might specify configuration settings which supercede the values in a baseline.

5.1 Implement the Baseline

Baseline Implementation is the process of configuring appropriate devices in the field to be consistent with the baseline. When a baseline is updated, older versions of that baseline represent allowable configurations for an applicable device for some amount of time.

The work of performing updates is not in the scope of this document.

5.2 Verify the Baseline

Baseline verification is an assurance process where applicable devices for a baseline are identified, and the devices themselves are audited to see that settings in that baseline have been applied. How a specific setting is verified depends on the ease of verification. Some items can be routinely verified by software agents on a computer. Other items, for example physical security can only be identified by physical inspection. The verification process is sensitive to the fact that:

- A CE can be configured to any valid baseline.
- A CE may have a variance allowing that certain settings specified in the baseline need not be made. Verification SHALL ignore these settings.
- A CE may have a variance specifying alternate settings. These settings SHALL be verified

5.3 Announce Baseline Changes

Because baselines can incorporate dynamic settings, (for example specifications of patch levels), a given computer may come out of compliance even though the baseline document has not been updated or deprecated.

Announce Baseline Changes is a process that communicates to the administrators that an **indirect** configuration setting will soon become invalid. Announcements are made in

some TBD reasonable manner, and TBD reasonable frequency. The announcement process augments the fundamental responsibility of a system administrator to know and follow applicable baselines for the system.

5.4 Enforce a Baseline

Baseline Enforcement is a process that analyzes and acts on systems identified by the verification process as being out of compliance. The spirit of the enforcement process is to notify the party responsible for the configuration, and to provide for a period of remediation.

During the remediation period, Enforcement notifies system managers in some reasonable manner and some reasonable frequency. Recall that the duration of the remediation period is configuration-item specific, and specified in the baseline. Recall that for configuration items of grave importance, the period of remediation can be 0.

Remediation can occur by either configuring the system according to the baseline, or by obtaining a variance.

When the period of remediation expires and the system remains out of compliance, the enforcement action is taken. The enforcement action is a configuration-item specific action. In unusual cases, the manager may grant relief beyond the period of remediation specified in the baseline.